(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 March 2002 (07.03.2002)

PCT

(10) International Publication Number
WO 02/19069 A2

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number: PCT/GB01/03852

(22) International Filing Date: 29 August 2001 (29.08.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0021444.5 31 August 2000 (31.08.2000) GB

(71) Applicant *(for all designated States except US)*: CONTENT TECHNOLOGIES LIMITED [GB/GB]; 1310 Waterside, Arlington Business Park, Theale, Reading, Berkshire RG7 4SA (GB).

(72) Inventor; and

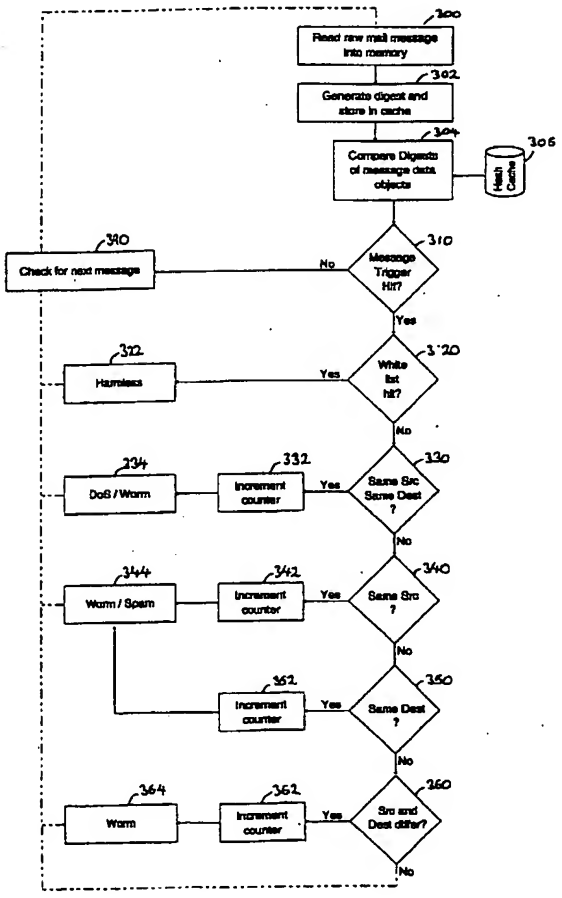
(75) Inventor/Applicant *(for US only)*: HOCKEY, Alyn [GB/GB]; c/o Content Technologies Limited, 1310 Waterside, Arlington Business Park, Theale, Reading, Berkshire RG7 4SA (GB).

(74) Agent: O'CONNELL, David, Christoper; Haseltine Lake & Co., Imperial House, 15-19 Kingsway, London WC2B 6UD (GB).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AT (utility model), AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, CZ (utility model), DE, DE (utility model), DK, DK (utility model), DM, DZ, EC, EE, EE (utility model), ES, FI, FI (utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (utility model), SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

[Continued on next page]

(54) Title: MONITORING ELECTRONIC MAIL MESSAGE DIGESTS



(57) Abstract: A method for monitoring electronic mail messages, each mail message comprising header information and a main body, particularly for protection against virus attacks and unsolicited commercial email (UCE). The method comprises generating a summary digest of only the subject line and the message content of the main body, wherein the message content may comprise textual content and/or attached files. The generated summary digest is stored in a memory, and compared with existing summary digests stored in memory. If the number of matches exceeds a threshold value, an alert signal is raised and appropriate action initiated. A timestamp may be stored with each summary digest, together with sender/recipient details and the internet protocol (IP) address of origin, to aid detection of the originator of the message.

WO 02/19069 A2

(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Monitoring electronic mail message digests

Technical Field of the Invention

This invention relates to networked computer systems security in general and protection against Denial of Service (DoS) attacks, virus attacks and unsolicited commercial email in particular. More specifically, this invention concerns an apparatus and method for managing electronic mail message processing.

Background to the Invention

Recent years have witnessed a proliferation in the use of the Internet. Many stand-alone computers and local area networks connect to the Internet for exchanging various items of information and/or communicating with other networks.

Such systems are advantageous in that they can exchange a wide variety of different items of information at a low cost with servers and networks on the Internet and other networks.

However, the inherent accessibility of the Internet increases the vulnerability of a system to threats such as cracker attacks, Denial of Service (DoS) attacks, viruses and unsolicited commercial email (UCE). Around 5-10 new viruses are discovered each day on the popular Windows-based operating systems. Especially insidious are those that propagate through the Internet, for example by using mail messages as a transport mechanism, known as email worms. The concern for advanced security solutions for networked systems in particular is therefore substantial.

In contrast to traditional viruses which are designed to spread themselves on a single computer using the file handling capabilities of an operating system, a worm exploits a computer's networking capabilities as the transport mechanism to enable it to infect other machines. Email worms, for instance, may comprise a file attachment to a mail message, or comprise a script or code embedded into the text of a mail message. The worm may exploit scripting capabilities of internet mail client software to send malicious code to other users on a mailing list or newsgroup automatically, and thus appear as having originated from a legitimate sender.

In particular, an email worm may exploit a user's email address book to obtain targets to spread to. To the recipient, the worm would appear to have been sent by a familiar source, as their presence in the address book would imply that previous email dialogue had taken place. Therefore, a further email apparently from that source would not appear out of the ordinary. Such emails may comprise a suitably benign message as further means of deception to the recipients.

Especially threatening are email worms which require no action on the part of the recipient (such as opening a mail message) to install and activate the malicious code. For instance, some mail clients, such as Outlook Express (™) and Outlook (™) support a "preview" pane which typically scans all unread messages in a user's electronic mailbox. Such a "preview" is sufficient to install and activate such a worm if present amongst the unread messages.

Other types of worms are known. For instance, a worm could spread over a local area network (LAN), wide

area network (WAN) or peer-to-peer network directly by determining the network addresses of other computers on the network and sending copies of its code to these addresses. Such files would appear to originate from a legitimate machine.

A further category of worm spreads using internet relay chat (IRC), a protocol allowing real time communications between Internet users and other features such as transfer of data and executable files over IRC channels. Popular IRC client software supports scripting features. For example, an IRC script could send a message or data file automatically to specified users when they connect to an IRC channel. Special scripting commands allow execution of DOS and Windows executable files allowing infected scripts to propagate and transfer worm code to other machines.

A growing concern amongst Internet users is that of unsolicited commercial email (UCE), commonly referred to as "spam". The ease and low cost of distributing email messages has made mass marketing via email an attractive advertising medium, particularly for bogus homeworking schemes, business and investment opportunities, lottery schemes, money-making clubs and chain letters. Bogus schemes are often characterised by exaggerated earnings claims, glowing testimonials, "no risk" guarantees and legal assurances. Such "spam" messages have led to added costs for both recipients and internet service providers (ISPs) in the form of additional bandwidth, disk space, server resources, and lost productivity. Furthermore, many users consider "spam" messages offensive and an invasion of privacy. The current growth rate of "spam" suggests that the problem may become unmanageable if it continues to grow at the current rate.

A known approach for addressing the problem of "spam" is to use a mail filtering system implemented at an Internet Service Provider (ISP) server, organisation's mail transfer agent (MTA) or user terminal. Such a filtering system sorts incoming email into categories, typically determined by the recipient. WO 99/32985 discloses a system and method of filtering junk emails comprising a first list of unapproved email addresses and character strings which a user wishes not to receive and a second list of approved email addresses and character strings which the user wishes to receive. The first list is periodically updated on the basis of further mail messages that the user rejects. However to remain effective, the filtering instructions must frequently be updated as new junk mail messages appear. Furthermore, it does not aid in tracking the originator of "spam" messages.

EP 0720333A discloses a technique for reducing the quantity of "spam" messages received by a user whereby a recipient specifier containing non-address information is added to an email message. The mail filter for a given recipient has access to information about that recipient and uses that information together with the non-address information in the email message to determine whether the message should be provided to the given recipient. If the non-address information and the information about the recipient indicate that the given recipient should not receive the message, the filter does not provide it. Such "non-address information", however, comprises two additional components, namely a recipient specifier and a referral list, which accompany the email message. These components must be adopted by all parties wishing to communicate with the specified user for the method to be effective. Furthermore, said information increases

the size of the mail message and hence storage space and bandwidth requirements. In addition, this method is of limited use in detecting email worms as email worms are characterised by their traffic behaviour rather than by any particular sender.

In the method of WO 99/33188, a system is provided for controlling delivery of unsolicited electronic mail messages, whereby "spam" probe email addresses are created and planted at various sites on a communications network in order to ensure their inclusion on large-scale "spam" mailing lists. Upon receipt of incoming email addressed to the spam probe addresses, the received email is analysed automatically to identify the source of the message, characteristic spam source data extracted from the message, and an alert signal generated containing the spam source data. A filtering system implemented at the servers receives the alert signal and updates filtering data using the spam source data retrieved from the alert signal, and controls delivery of subsequently received email messages received from the identified spam source.

One shortcoming of this method is that, whilst convenient for monitoring spam, it is of limited utility against a DoS attack, email worm or other traffic based mechanism, such as an incorrectly configured mail system that accidentally loops messages.

Accordingly, there is a need for a system that automatically and efficiently identifies unsolicited or threatening email messages and controls the delivery of these messages to users, for example by preventing delivery of these messages, or by identifying the messages as unsolicited by displaying the messages in a distinctive display mode.

Statement of the Invention

It is an object of the present invention to provide a system and a computer-implemented method for automatically and efficiently identifying recurrent mail messages. Such a method is particularly advantageous for users of electronic mail, for example, in filtering email worms and unsolicited commercial email (UCE).

According to a first aspect of the present invention, there is provided a method for monitoring electronic mail messages, wherein each mail message may comprise:

(a) subject header information

(b) non-subject header information

(c) a main body, comprising message content;

the method comprising:

receiving an electronic mail message,

generating a characteristic numerical representation of at least a part of said message content, but not of said non-subject header information;

storing said generated characteristic numerical representation in a memory; and

comparing said characteristic numerical representation with each characteristic numerical representation stored in memory.

Such a method is advantageous as a large quantity of data may be rapidly and automatically searched for matches, whilst significantly reducing the amount of data to be stored. When implemented, for example, in a mail server or organisation's mail transfer agent (MTA), a considerable database can be assembled over a time period. This ensures that the searching is extensive and also enables monitoring of traffic activity.

Preferably, the step of generating comprises generating a characteristic numerical representation of at least a part of said message content and of said subject header information.

Preferably, the step of generating a characteristic numerical representation of at least a part of said message content comprises generating a characteristic numerical representation of at least textual content in said message content.

Preferably, the step of generating a characteristic numerical representation of at least a part of said message content comprises generating a characteristic numerical representation of at least attached files in said message content. Generating characteristic numerical representations of attached files assists in monitoring of email worms which rely on file attachments in order to propagate. In an embodiment, the step of generating may produce a characteristic numerical representation of said file attachments only. This enables the speed of the search to be increased, for example, when scanning for virus worms.

Preferably, a timestamp of said electronic mail message is stored. Such a timestamp may be provided from a defined public time service, such as a public time server on the Internet. Preferably, the internet protocol (IP) address of origin of said electronic mail message is further stored. These assist in detecting of the originator of said mail message.

Preferably, the header information is stored. Further preferably, said header information comprises message source and destination details. Comparison of

these allows determination of whether the message is a simple resend, a bulk circulated email, or transmitted via multiple routes.

Preferably, said generated characteristic numerical representation is a message digest, for example generated using a message digest 5 (MD5) algorithm or a hash, for example generated using secure hash algorithm 1 (SHA-1). These algorithms produce a condensed, fixed size hash of 128 bits and 160 bits respectively, and are essentially unique, permitting a fast and straightforward comparison of the electronic mail messages to be made.

Preferably, said method further comprises, if said characteristic numerical representation matches a characteristic numerical representation stored in memory, incrementing a count value associated with said characteristic numerical representation.

Further preferably, said method further comprises comparing said count value with a predetermined threshold. Still further preferably, said predetermined threshold is determined on the basis of said header information. This enables a different threshold to be set for characteristic numerical representations having the same sender and destination information, a same sender and a different destination, a different sender but a same destination, and different senders and destinations.

If said count value exceeds said predetermined threshold, an alert is preferably raised. Furthermore, said alert is preferably determined on the basis of said header information.

Preferably, said alert comprises flagging said message with a marker prefixed to the subject line of the message. In an embodiment, said marker may comprise the word "JUNK", "SUSPICIOUS" and may further be user configurable.

Preferably, said alert comprises delivering a fixed portion of said message only. For example, the first 512 and last 512 bytes of a message could be delivered to the recipient. Preferably, said alert comprises delivering an audit notification to the recipient. This would enable the recipient to confirm that the message is unsolicited whilst economising on bandwidth.

Preferably, said alert comprises deleting the message. Preferably, said alert comprises deleting a file attachment from the message. This facility is useful in instances where the message has already been sent to a same destination a number of times, where a message is suspected of being an email worm, or in guarding a destination or mail server thereof against cracker attack.

In a preferred embodiment, said method further comprises, allowing said message to be delivered if said characteristic numerical representation matches an approved characteristic numerical representation stored in memory. Such a "white list" of approved characteristic numerical re-presentations may include those of routine test messages, or other routinely sent "bulk" messages.

Preferably, said characteristic numerical representation is stored in cache memory. Further preferably, the maximum number of characteristic numerical repre-

sentations stored in cache memory is configurable and said cache memory may preferably be periodically copied to a storage means. The use of cache memory to store recent characteristic numerical representations allows the database of characteristic numerical representations to be searched more quickly.

In accordance with a second aspect of the present invention, there is provided a software product for use in a server or organisation's mail transfer agent (MTA), for monitoring electronic mail messages, wherein each mail message may comprise:

(a) subject header information

(b) non-subject header information

(c) a main body, comprising message content;

the software containing code for:

receiving an electronic mail message,

generating a characteristic numerical representation of at least a part of said message content, but not of said non-subject header information;

storing said generated characteristic numerical representation in a memory; and

comparing said characteristic numerical representation with each characteristic numerical representation stored in memory.

In accordance with a third aspect of the present invention, there is provided a software product for use in a client's mail user agent (UA) for monitoring electronic mail messages, wherein each mail message may comprise:

(a) subject header information

(b) non-subject header information

(c) a main body, comprising message content;

the software product containing code for:

receiving an electronic mail message,

generating a characteristic numerical representation of at least a part of said message content, but not of said non-subject header information;

storing said generated characteristic numerical representation in a memory; and

comparing said characteristic numerical representation with each characteristic numerical representation stored in memory.

In accordance with a fourth aspect of the present invention, there is provided a computer system for monitoring electronic mail messages, wherein each mail message may comprise:

(a) subject header information

(b) non-subject header information

(c) a main body, comprising message content;

the system comprising:

means for receiving an electronic mail message,

a processor for generating a characteristic numerical representation of at least a part of said message content, but not of said non-subject header information;

a memory for storing said generated characteristic numerical representation; and

means for comparing said characteristic numerical representation with each characteristic numerical representation stored in memory.

Brief description of the drawings

For a better understanding of the present invention, and to show more clearly how it may be carried into effect, reference will now be made, by way of example, to the accompanying drawings, in which:-

Figure 1 is a block diagram of part of a computer

network operating in accordance with the invention.

Figure 2 is a multi-part email message illustrating message body structure.

Figure 3 is a flowchart illustrating operation of a software product in accordance with the invention.

Detailed description of the preferred embodiments of the invention

Figure 1 of the accompanying drawings illustrates functional blocks of a server 100, such as a simple mail transfer protocol (SMTP) server, operable in accordance with the present invention. Server 100 comprises a central processing unit (CPU) 102 in communication with a memory 104. The CPU 102 can store and retrieve data to and from a storage means 106, and outputs display information to a video display 108.

Server 100 may be connected to and communicate with a private network 110 such as a local area network (LAN). In addition, server 100 may be able to send and receive files to and from a public network 116 such as the Internet, using an ISDN, serial, Ethernet or other connection, preferably via a firewall 112 and router 114. Internet 116 comprises a vast number of computers and computer networks that are connected through communications links.

Alternatively, local area network 110 may itself be connected through a server to another network (not shown) such as the Internet.

Server 100 may further comprise input peripherals such as a terminal having a mouse and/or keyboard (not shown) and output peripherals such as a printer or sound

generation hardware, as customary in the art. Server 100 runs operating system and networking software which may be stored on disc or provided in read-only memory (ROM). Data may be transferred to server 100 via a removable storage means (not shown) or through either of networks 110, 112.

One format of data that may be transferred between servers is electronic mail (email) messages, typically using the Simple Mail Transfer Protocol (SMTP). A server adapted for this purpose may also be known as a Message Transport System (MTS). A variety of mail server software implementing the SMTP and other protocols is available for popular hardware and operating systems.

Mail reader software, also known as Mail User Agents (MUA), allows end users, such as customers which connect to their ISP using a dial-up modem connection, to access and read their email, typically using the Post Office Protocol (POP3) or Internet Message Access Protocol (IMAP). In addition, messages may be sent, typically using the SMTP protocol.

The accepted standard format for messages carried by the Internet mail system is defined by Request for Comments (RFC) 822 (published on the Internet). As shown in Figure 2, email message 200 comprises header information 210, including subject header information 215, typically input by the sender of the message, and non-subject header information, including the identity of the sender and the intended recipient(s), and the date. The header information is followed by the body 230 of the message, conventionally separated from the header by a blank line 220.

The order of header information 210 is not important

in most cases, and many headers 214 are optional, but the mandatory headers 212 that must be present are Date:, From: and one of To:, Cc: or BCC:. The keyword may be in any mixture of capitals and small letters, so CC: is the same as Cc:. The subject header 215 although optional, is generally used.

The main body 230 of the message may comprise plain American Standard Code for Information Interchange (ASCII) text, or may assume a multi-part format allowing textual and non-textual message bodies to be represented and exchanged without loss of information. For instance, the Multipurpose Internet Mail Extensions (MIME) standard was created to address limitations concerning the structure and content of an email. Such a multi-part format, defined by RFC2045 and RFC2046 (published on the Internet), also permits representation of body text in character sets other than ASCII through content encodings, as well as representation of non-textual material such as images and audio fragments.

Finally, the message body may be appended by an optional "signature", comprising a signature separator 300 followed by signature text 310. Conventionally, the signature separator 300 comprises two hyphens followed by a space but often may comprise a series of hyphens, asterisks or other characters. The signature text 310 typically includes a user's name, organisation, and contact details.

The structure of a multi-part message will now be described in detail. As Figure 2 shows, further fields 216 relating to multi-part information are added to the header 210. Multi-part headers 216 identify the message 200 as being in multi-part format, and are used by mail

reader software to interpret the message 200 correctly. The fields 216 comprise distinctive boundary separator text 218 which is used to separate different parts of the message body 230 pertaining, for example, to textual content, enriched textual content such as HyperText Markup Language (HTML), and attached files. As an example, assuming a file attachment is present, field 216 may read:

Content-Type: multipart/mixed; boundary="ABCDEFG"

where "ABCDEFG" is the distinctive separator text 218.

Alternatively, the header may read

Content-Type: multipart/alternative; boundary="ABCDEFG"

the use of the word "alternative" indicating by convention that the same content is repeated in different parts of the message but enhanced in some of the repetitions (e.g. message text in both plain and HTML versions).

The different parts of the message body 230 are separated by boundary separators 240, 260 comprising two hyphens followed by the boundary separator text 218. Although body 230 is shown as consisting of two parts 250 and 270, email message 200 may comprise more or fewer parts.

Any preamble text 232 positioned above the first boundary separator 240 will typically be ignored by a mail reader that understands multi-part format messages but will be displayed by a mail reader which does not

support the multi-part format. That is to say, preamble text 232 is not considered part of the message content. Conventionally, such preamble text 232 may therefore be used to alert the recipient to the fact that the mail reader being used is not compliant with the format of the message, as shown.

The structure of the body parts 250, 270 will now be described, again with reference to Figure 2. First body part 250 encloses header fields 252 and a body 256 which is separated from header 252 by a blank line 254. Header fields 252 conventionally define the type of data contained in body 256 and may specify any encoding necessary for the purposes of transfer of information contained in the body 256 through the mail system, so that the receiving program can reverse the process. A file conveyed within a message part body 256 in this way is known as an "attachment" to the message. Alternatively, header fields 252 may be empty, in which instance the body is conventionally interpreted as comprising US-ASCII text.

The structure of second body part 270, and any further body parts, is analogous to body part 220, and the reference numerals indicate like parts. Further body parts may then follow as generally indicated by 280.

To indicate the end of the multi-part body 230, the body 230 is terminated by an end of body separator text 290. End of body separator text 290 comprises two hyphens followed by the distinctive boundary separator text 218, followed by a further two hyphens.

Reference will now be made to Figure 3, which describes the operation of an embodiment of the software

in accordance with the invention. Preferably, the software is loaded permanently on a mail server, and remains in a quiescent state until an email message arrives at the server, typically from another server via SMTP. In step 300, the software intercepts an incoming mail message and reads it into memory. The software then determines whether the message is in multi-part format, and, if so, whether there are any file attachments to the message. Typically, this is done by examining the header for the following fields as part of step 300

MIME-Version
and Content-Type: multipart/xxx

where xxx may indicate a number of alternatives as customary in the art, such as "mixed", "alternative" and "parallel".

To determine whether the parts correspond to attached files, the body of the message may be scanned for the header field:

Content-Type: xxx

xxx will typically read "Text/plain", "Text/html" or similar, for a message without any non-textual file attachments. If file attachments are present, xxx may read "image/jpeg", "audio/basic" or a number of other possibilities corresponding to a variety of file formats as customary in the art. The header would typically read "Content-Type: multipart/mixed" accordingly.

A characteristic numerical representation is generated for the combined subject line 215 and message content, in step 302. In the multi-part example given,

the message content is obtained by taking the body 256 of the first part (text) 250 and the body 276 of the second part (image) 270, that is to say, any multi-part boundary information and any header information of each part is not considered to be part of the message content. Alternatively, if the message were not in the multi-part MIME format, the message content would be considered to be the entire body 230 of the message 200.

Such a characteristic numerical representation, or "hash" as known to one of skill of the art, is a message digest algorithm which takes a message of arbitrary length and produces a numerical representation comprising a number of bits sufficiently small to form a condensed digest of the original message and allow fast and straight-forward searching, but sufficiently large to be essentially unique.

Example algorithms include Message Digest 5 (MD5), developed by Rivest in 1991 and documented in Internet RFC1321 (published on the Internet) which takes a message of arbitrary length and produces a 128-bit message digest. An alternative is the Secure Hash Algorithm 1 (SHA-1) developed by National Institute of Science and Technology (NIST) which produces a 160-bit message digest. In both these cases, the messages are padded so they are a multiple of 512 bits long and processed in blocks if necessary.

Although in this example, a single digest has been generated for a multi-part message, in an alternative, variations such as generating a first digest for the subject only, a second digest for the message content or separate digests for each body part may be performed. A user, such as a System Administrator, may define which

elements would be used in generating the digests.

However, since a characteristic of virus worms is that they may cause the same message to be sent from different sources to different users in the network, it is preferable not to use the non-subject header information in generating the digest, if such a virus is to be detected.

The generated digest (or digests) is stored in a memory, together with the internet protocol (IP) address of where the message came from, sender information and destination details, and a timestamp to serve as a record of when the message was received. For instance, the time from a defined public time service, such as a public time server on the internet, may be used. Other information from the header may be stored such as the Message-ID field. Again, these options may be configured by a System Administrator.

To determine whether the textual content of the message or the attached files match those from previous messages received by the mail server 100, the digest is compared with existing digests stored in memory in step 304 and any matches noted in step 310. To this end, the use of message digest algorithms is advantageous in that they greatly reduce the amount of data to be stored and hence enable fast searching, whilst remaining essentially unique and thus ensuring the probability of conflict with a different mail message is extremely low. These considerations are important given that a typical user may be sent between 20-40 emails on average over a single 24 hour period.

For the method to be most effective, ideally the

digest should be compared with those of mail messages sent over approximately the previous seven days, as typical spam messages will recur within a short period. Taking an organisation with 8000 users as an example, in which approximately one quarter of the emails sent share common content and hence produce a same digest, the total minimum number of digests to be stored at any one time is therefore in the order of 7 days × 30 emails/day × 8000 users × 0.75 = 1,260,000. To accelerate the process, these most recent digests, in this case, those of mail messages sent over the last seven days, are stored in a cache memory 306 which could be flushed to disk periodically.

If the digest does not match an existing digest stored in cache memory 306, a check is performed in step 390 to determine whether there are further messages waiting to be processed by the mail server, and if so, the process will loop back to step 300 and read the next message in to memory. If no more messages are waiting, the process returns to a quiescent state in which it may perform background tasks such as transferring older hashes out of cache memory and onto disk whilst waiting for further messages to arrive.

Alternatively, if the digest matches an existing digest stored in memory, the following operations are performed:

Initially, the digest is compared in step 320 against a "white list" of approved digests believed to be harmless. These typically include test messages sent routinely, particularly by new users, to determine whether a mail system is functioning correctly. If the message is found to match such a digest, as indicated by

322, the message is allowed to proceed and the process will loop back to step 300 accordingly.

Alternatively, if the digest matched is not an "approved" digest, the sender/destination information (as indicated by the From:, To:, Cc: and Bcc: headers) and/or other header information stored with those hashes are compared. Firstly, in step 330, a check is performed to see if the digests share a common source and destination, and if so, a count value for this instance is incremented in step 332. One or more of a variety of actions may be performed, depending on the sender/recipient information and on how large a count value has accrued for the hash in question.

The most basic action possible is to allow the message to continue. This is appropriate, if the tally lies below a small threshold value (e.g. 5), as the sender and recipient details are the same. Such an instance could correspond to a simple resend.

However, if the threshold is exceeded, but still lies below a second value (e.g. 20), the message could be flagged as unsolicited, e.g. by displaying the message with a "Suspicious" or "Junk" prefix in the subject line on the video display 110, and transmitting only the header information, to save the recipient from downloading the message. Again, the thresholds and attributes used for distinguishing a flagged message may be user-configurable.

Finally, if this second threshold is exceeded, it is likely that the message could be a mailbomb or a form of cracker attack, such as the Denial of Service (DoS). In this case, the message could be deleted at the server.

A second possibility is that the digests may share a common source but have different destination information as indicated by 330. Again, a counter is incremented in step 342. Different thresholds and actions would correspondingly apply in this instance. As a variety of recipients are involved, the lowest threshold in this case may be set at a considerably higher value, such as 100, as a user may legitimately send a same message to a variety of recipients, such as a business newsletter, party invitation or information concerning a change of address. However, it is more conventional for a user to do this by specifying multiple destinations in the header information rather than to send separate messages each containing the same information.

If this threshold is exceeded, the message could be a spam message or an email worm, as indicated by 344. As well as flagging a message as suspicious, a condensed form of the message could be safely delivered to the recipient (comprising the first and last 512 bytes of the message by way of example), together with an audit notification to inform the recipient that the message is suspected to be unsolicited. This action could inhibit harmful behaviour of the worm whilst enabling the user to verify whether the message is unsolicited and to request the entire message, if necessary.

In yet a further possibility, a message could share a common recipient as indicated by 350. Again, a counter for this instance is incremented in step 352. Such a situation could also correspond to both spam or worm activity.

In still yet a further alternative, a message may share a variety of senders and recipients. The threshold for taking the action of deleting the message may be set very low (e.g. 40). Such a message is likely to be an email worm, especially when the subject line is found to match in most cases and the majority of timestamps are found to lie within a very short period, such as seven days or less. A scoring system may be used to lower the threshold required if the file attachments, if any, are found to correspond, or the subject line is found to match.

It will be understood that this aspect of the process is subject to variations as customary in the art. For example, other options include changing the message attributes so that it may not be delivered or opened other than by a system administrator, and/or may place the file in a "quarantine zone"; an area of filespace with restricted access for review by a system administrator. Such quarantine zones are largely conventional in the art, e.g. used by junk and spam mail filtering programs to filter mail which is thought to be unsolicited. Typically, these options and thresholds would remain configurable by a system administrator.

A further application of the present invention is in tracking the originator of a spam message or worm. A system operator could perform a search of all messages that matched the "hash" value, to determine who sent the first message and where it originated. For instance, the internet protocol (IP) address stored with each message could be used to provide this information. Depending on the number of "fields" stored with the digest, more detailed information could be obtained. Appropriate action could then be taken to identify the perpetrator.

Such a method is more effective than simply scanning the header fields in a single spam message, as these can be substituted by fraudulent header fields, often by experienced perpetrators of spam messages. A typical example is the substitution of the "received:" header.

There is thus described a method, software product and a computer system which provide for detecting email worms and spam messages.

Although the software is primarily intended for use in a mail server of an Internet Service Provider (ISP), or in an organisation's mail transfer agent (MTA), the software could also be used at the client end, or mail user agent (MUA). Such software would function on an identical principle. However, it is preferable for the software to run on a mail server or MTA for the following reasons:

(1) the software controls delivery of the messages, and therefore is able to maximise use of a client's bandwidth to transferring legitimate messages.

(2) the database of previous digests to compare with is much larger than available on a client's machine, and so there is a relatively high chance of spotting a spam message or email worm.

The ability to run the software on a end user's machine is of use, however, where the user's ISP does not run software of this type, or the user has a need to download mail from several POP3 or IMAP accounts.

Yet a further extension of the above method is to use the software to maintain a store of digests for outgoing messages. The software could alert a user if he or she is suspected of circulating spam messages, or file

attachments which may include a virus worm.

It is noted that the various options described above may be programmed or configured by a user and that the above detailed description of preferred embodiments of the invention is provided by way of example only. Other modifications which are obvious to a person skilled in the art may be made without departing from the true scope of the invention, as defined in the appended claims.

CLAIMS

1. A method for monitoring electronic mail messages, wherein each mail message may comprise:

(a) subject header information

(b) non-subject header information

(c) a main body, comprising message content;

the method comprising:

receiving an electronic mail message,

generating a characteristic numerical representation of at least a part of said message content, but not of said non-subject header information;

storing said generated characteristic numerical representation in a memory; and

comparing said characteristic numerical representation with each characteristic numerical representation stored in memory.

2. A method for monitoring electronic mail messages as claimed in claim 1, in which said step of generating comprises generating a characteristic numerical representation of at least a part of said message content and of said subject header information.

3. A method for monitoring electronic mail messages as claimed in claims 1 or 2, in which said step of generating a characteristic numerical representation of at least a part of said message content comprises generating a characteristic numerical representation of at least textual content in said message content.

4. A method for monitoring electronic mail messages as claimed in any preceding claim, in which said step of generating a characteristic numerical representation of at least a part of said message content comprises

generating a characteristic numerical representation of at least attached files in said message content.

5. A method for monitoring electronic mail messages as claimed in any preceding claim, in which the step of storing further comprises storing a timestamp of said electronic mail message.

6. A method for monitoring electronic mail messages as claimed in any preceding claim, in which the step of storing further comprises storing the internet protocol (IP) address of origin of said electronic mail message.

7. A method for monitoring electronic mail messages as claimed in any preceding claim, in which said step of storing further comprises storing header information of said electronic mail message.

8. A method for monitoring electronic mail messages as claimed in claim 7, in which said step of storing further comprises storing message source and destination details.

9. A method for monitoring electronic mail messages as claimed in any preceding claim, in which the characteristic numerical representation is a message digest.

10. A method for monitoring electronic mail messages as claimed in claim 9, in which said message digest is generated using a message digest 5 (MD5) algorithm.

11. A method for monitoring electronic mail messages as claimed in claim 9, in which said message digest is generated using a secure hash algorithm 1 (SHA-1) algorithm.

12. A method for monitoring electronic mail messages as claimed in any preceding claim, further comprising, if said characteristic numerical representation matches a characteristic numerical representation stored in memory, the step of incrementing a count value associated with said characteristic numerical representation.

13. A method for monitoring electronic mail messages as claimed in claim 12, further comprising the step of comparing said count value with a predetermined threshold.

14. A method for monitoring electronic mail messages as claimed in claim 13, in which said predetermined threshold is determined on the basis of said header information.

15. A method for monitoring electronic mail messages as claimed in claims 13 or 14, further comprising, if said count value exceeds said predetermined threshold, raising an alert.

16. A method for monitoring electronic mail messages as claimed in claim 15, in which said alert is determined on the basis of said header information.

17. A method for monitoring electronic mail messages as claimed in claims 15 or 16, in which said alert comprises a marker prefixed to the subject line of the message.

18. A method for monitoring electronic mail messages as claimed in claims 15 to 17, in which said alert comprises delivering a fixed portion of said message.

19. A method for monitoring electronic mail messages as claimed in claims 15 to 18, in which said alert comprises delivering an audit notification to the recipient.

20. A method for monitoring electronic mail messages as claimed in claims 15 to 19, in which said alert comprises deleting said message.

21. A method for monitoring electronic mail messages as claimed in claims 15 to 19, in which said alert comprises deleting a file attachment from the message.

22. A method for monitoring electronic mail messages as claimed in any preceding claim, comprising allowing said message to be delivered if said characteristic numerical representation matches an approved characteristic numerical representation stored in memory.

23. A method for monitoring electronic mail messages as claimed in any preceding claim, in which said characteristic numerical representation is stored in cache memory.

24. A method for monitoring electronic mail messages as claimed in claim 23, in which said cache memory may be periodically copied to a storage means.

25. A software product for use in a server or organisation's mail transfer agent (MTA), for monitoring electronic mail messages, wherein each mail message may comprise:

(a) subject header information

(b) non-subject header information

(c) a main body, comprising message content;

the software containing code for:

receiving an electronic mail message,

generating a characteristic numerical representation of at least a part of said message content, but not of said non-subject header information;

storing said generated characteristic numerical representation in a memory; and

comparing said characteristic numerical representation with each characteristic numerical representation stored in memory.

26. A software product for use in a client's mail user agent (UA), for monitoring electronic mail messages, wherein each mail message may comprise:

(a) subject header information

(b) non-subject header information

(c) a main body, comprising message content;

the software containing code for:

receiving an electronic mail message,

generating a characteristic numerical representation of at least a part of said message content, but not of said non-subject header information;

storing said generated characteristic numerical representation in a memory; and

comparing said characteristic numerical representation with each characteristic numerical representation stored in memory.

27. A software product as claimed in claims 25 or 26, in which said code for generating comprises code for generating a characteristic numerical representation of at least a part of said message content and of said subject header information.

28. A software product as claimed in any preceding claim, in which said code for generating a characteristic numerical representation of at least a part of said message content comprises code for generating a characteristic numerical representation of at least textual content in said message content.

29. A software product as claimed in any preceding claim, in which said code for generating a characteristic numerical representation of at least a part of said message content comprises code for generating a characteristic numerical representation of at least attached files in said message content.

30. A software product as claimed in any preceding claim, in which the code for storing further comprises code for storing a timestamp of said electronic mail message.

31. A software product as claimed in any preceding claim, in which the code for storing further comprises code for storing the internet protocol (IP) address of origin of said electronic mail message.

32. A software product as claimed in any preceding claim, in which said code for storing further comprises code for storing header information of said electronic mail message.

33. A software product as claimed in claim 32, in which said code for storing further comprises code for storing message source and destination details.

34. A software product as claimed in any preceding claim, in which the characteristic numerical representation is a message digest.

35. A software product as claimed in claim 34, in which said message digest is generated using a message digest 5 (MD5) algorithm.

36. A software product as claimed in claim 35, in which said message digest is generated using a secure hash algorithm 24 (SHA-1) algorithm.

37. A software product as claimed in any preceding claim, further comprising, if said characteristic numerical representation matches a characteristic numerical representation stored in memory, code for incrementing a count value associated with said characteristic numerical representation.

38. A software product as claimed in claim 37, further comprising code for comparing said count value with a predetermined threshold.

39. A software product as claimed in claim 37, in which said predetermined threshold is determined on the basis of said header information.

40. A software product as claimed in claims 38 or 39, further comprising, if said count value exceeds said predetermined threshold, code for raising an alert.

41. A software product as claimed in claim 40, in which said alert is determined on the basis of said header information.

42. A software product as claimed in claims 40 or 41, in which said alert comprises a marker prefixed to the subject line of the message.

43. A software product as claimed in claims 40 to 42, in which said alert comprises delivering a fixed portion of said message.

44. A software product as claimed in claims 40 to 43, in which said alert comprises delivering an audit notification to the recipient.

45. A software product as claimed in claims 40 to 44, in which said alert comprises deleting said message.

46. A software product as claimed in claims 40 to 44, in which said alert comprises deleting a file attachment from the message.

47. A software product as claimed in any preceding claim, comprising code for allowing said message to be delivered if said characteristic numerical representation matches an approved characteristic numerical representation stored in memory.

48. A software product as claimed in any preceding claim, in which said characteristic numerical representation is stored in cache memory.

49. A software product as claimed in claim 48, in which said cache memory may be periodically copied to a storage means.

50. A computer system for monitoring electronic mail messages, wherein each mail message may comprise:

(a) subject header information

(b) non-subject header information

(c) a main body, comprising message content;

the system containing:

means for receiving an electronic mail message,

means for generating a characteristic numerical representation of at least a part of said message content, but not of said non-subject header information;

a memory for storing said generated characteristic numerical representation; and

means for comparing said characteristic numerical representation with each characteristic numerical representation stored in memory.

51. A computer system for monitoring electronic mail messages as claimed in claim 50, in which said means for generating comprises means for generating a characteristic numerical representation of at least a part of said message content and of said subject header information.

52. A computer system for monitoring electronic mail messages as claimed in claims 50 or 51, in which said means for generating a characteristic numerical representation of at least a part of said message content comprises means for generating a characteristic numerical representation of at least textual content in said message content.

53. A computer system for monitoring electronic mail messages as claimed in claims 51 or 52, in which said means for generating a characteristic numerical representation of at least a part of said message content

comprises means for generating a characteristic numerical representation of at least attached files in said message content.

54. A computer system for monitoring electronic mail messages as claimed in any preceding claim, in which the means for storing further comprises means for storing a timestamp of said electronic mail message.

55. A computer system for monitoring electronic mail messages as claimed in any preceding claim, in which the means for storing further comprises means for storing the internet protocol (IP) address of origin of said electronic mail message.

56. A computer system for monitoring electronic mail messages as claimed in any preceding claim, in which said means for storing further comprises means for storing header information of said electronic mail message.

57. A computer system for monitoring electronic mail messages as claimed in claim 56, in which said means for storing further comprises means for storing message source and destination details.

58. A computer system for monitoring electronic mail messages as claimed in any preceding claim, in which the characteristic numerical representation is a message digest.

59. A computer system for monitoring electronic mail messages as claimed in claim 58, in which said message digest is generated using a message digest 5 (MD5) algorithm.

60. A computer system for monitoring electronic mail messages as claimed in claim 58, in which said message digest is generated using a secure hash algorithm 1 (SHA-1) algorithm.

61. A computer system for monitoring electronic mail messages as claimed in any preceding claim, further comprising, if said characteristic numerical representation matches a characteristic numerical representation stored in memory, means for incrementing a count value associated with said characteristic numerical representation.

62. A computer system for monitoring electronic mail messages as claimed in claim 61, further comprising means for comparing said count value with a predetermined threshold.

63. A computer system for monitoring electronic mail messages as claimed in claim 62, in which said predetermined threshold is determined on the basis of said header information.

64. A computer system for monitoring electronic mail messages as claimed in claims 62 or 63, further comprising, if said count value exceeds said predetermined threshold, means for raising an alert.

65. A computer system for monitoring electronic mail messages as claimed in claim 64, in which said alert is determined on the basis of said header information.

66. A computer system for monitoring electronic mail messages as claimed in claims 64 or 65, in which said alert comprises a marker prefixed to the subject line of the message.

67. A computer system for monitoring electronic mail messages as claimed in claims 64 to 66, in which said alert comprises means for delivering a fixed portion of said message.

68. A computer system for monitoring electronic mail messages as claimed in claims 64 to 67, in which said alert comprises delivering an audit notification to the recipient.

69. A computer system for monitoring electronic mail messages as claimed in claims 64 to 68, in which said alert comprises deleting said message.

70. A computer system for monitoring electronic mail messages as claimed in claims 64 to 68, in which said alert comprises deleting a file attachment from the message.

71. A computer system for monitoring electronic mail messages as claimed in any preceding claim, comprising means for allowing said message to be delivered if said characteristic numerical representation matches an approved characteristic numerical representation stored in memory.

72. A computer system for monitoring electronic mail messages as claimed in any preceding claim, in which said characteristic numerical representation is stored in cache memory.

73. A computer system for monitoring electronic mail messages as claimed in claim 72, in which said cache memory may be periodically copied to a storage means.

74. A method for monitoring electronic mail messages substantially as described herein with reference to Figure 3 of the accompanying drawings.

75. A software product for use in a server or organisation's mail transfer agent (MTA) for monitoring electronic mail messages substantially as described herein with reference to Figure 3 of the accompanying drawings.

76. A software product for use in mail user agent (UA) for monitoring electronic mail messages substantially as described herein with reference to Figure 3 of the accompanying drawings.

77. A computer system for monitoring electronic mail messages substantially as described herein with reference to Figure 3 of the accompanying drawings.

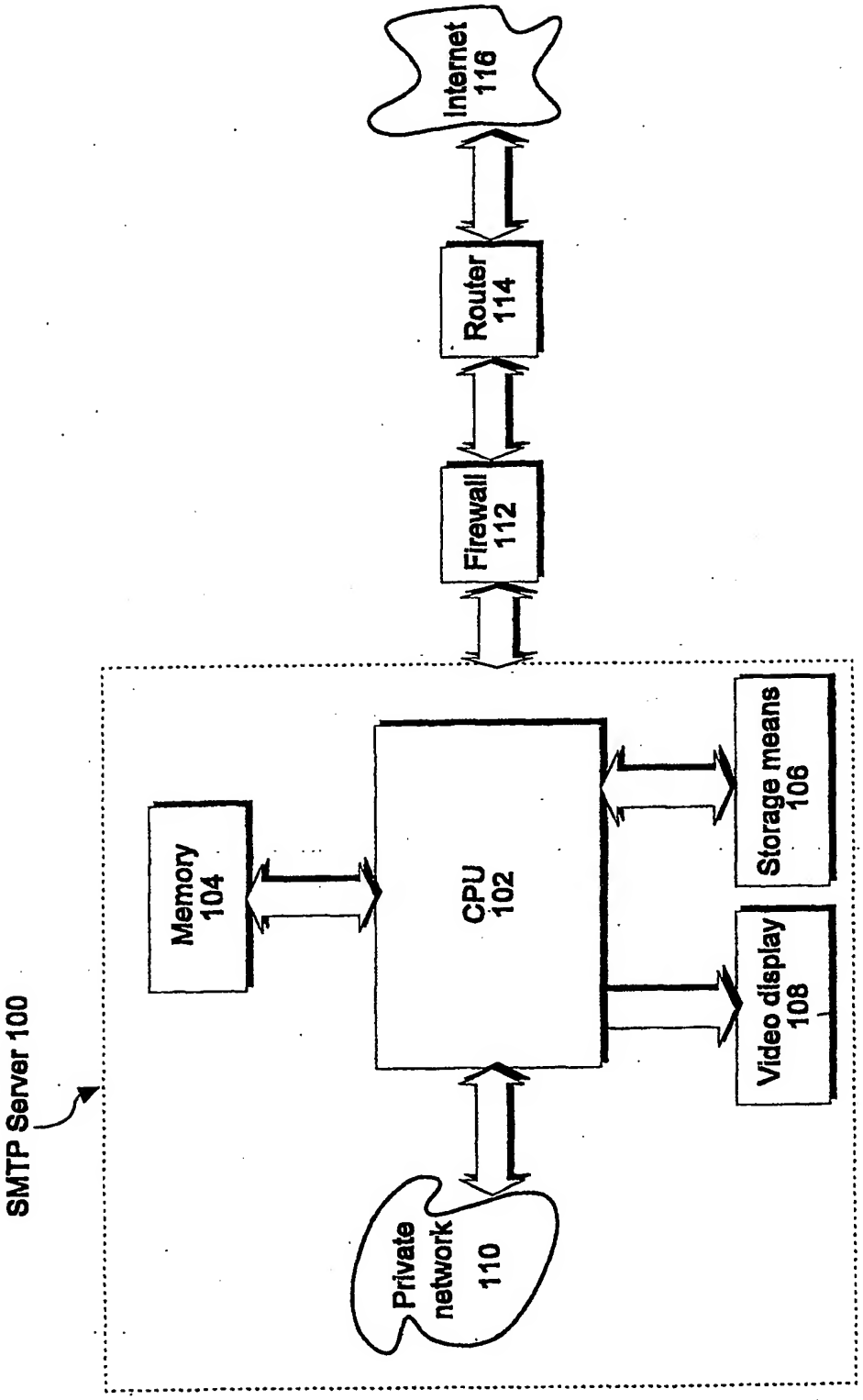

Internet 116
Router 114
Firewall 112
Memory 104
CPU 102
Storage means 106
Video display 108
Private network 110
SMTP Server 100

Figure 1

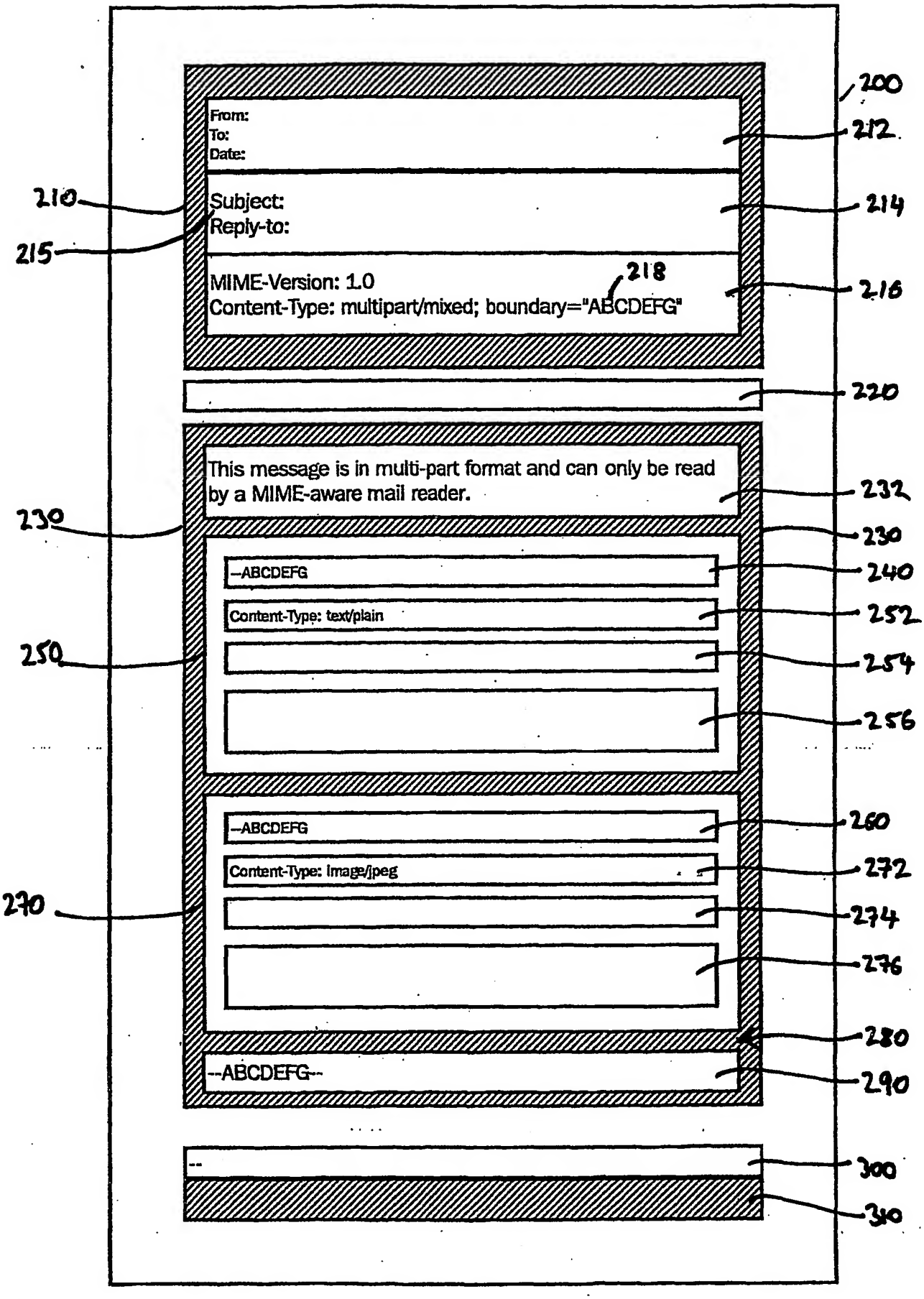

200
From:
To:
Date:
212
210
Subject:
214
215
Reply-to:
MIME-Version: 1.0
218
Content-Type: multipart/mixed; boundary="ABCDEFG"
216
220
This message is in multi-part format and can only be read by a MIME-aware mail reader.
232
230
230
--ABCDEFG
240
Content-Type: text/plain
252
250
254
256
--ABCDEFG
260
Content-Type: image/jpeg
272
270
274
276
280
--ABCDEFG--
290
--
300
310

Figure 2

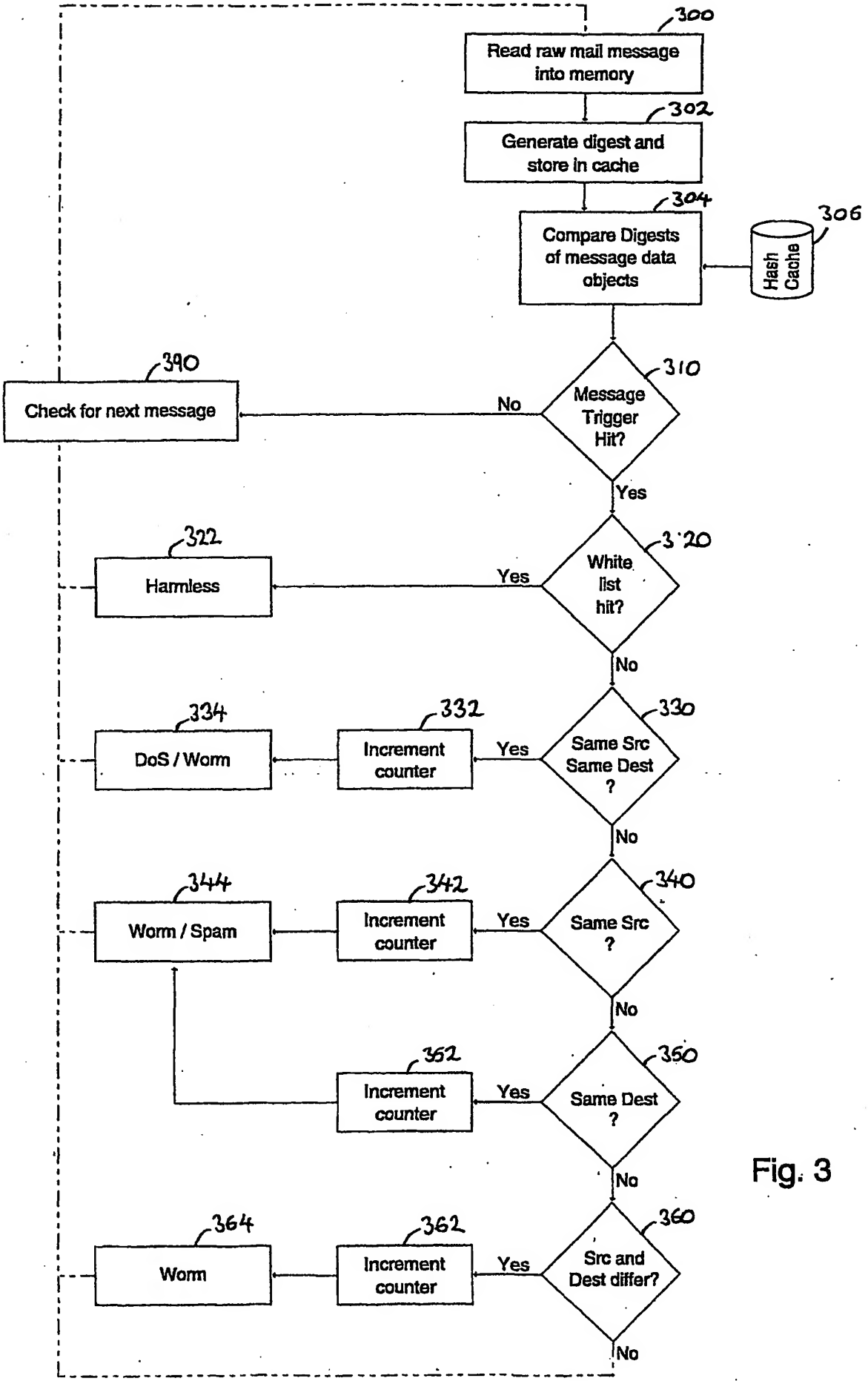

300
Read raw mail message into memory
302
Generate digest and store in cache
304
Compare Digests of message data objects
306
Hash Cache
310
Message Trigger Hit?
No
390
Check for next message
Yes
320
White list hit?
Yes
322
Harmless
No
330
Same Src Same Dest ?
Yes
332
Increment counter
334
DoS / Worm
No
340
Same Src ?
Yes
342
Increment counter
344
Worm / Spam
No
350
Same Dest ?
Yes
352
Increment counter
No
360
Src and Dest differ?
Yes
362
Increment counter
364
Worm
No

Fig. 3